

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-051845

(43)Date of publication of application : 23.02.2001

(51)Int.Cl.

G06F 9/38
G06F 9/305
G06F 9/32

(21)Application number : 11-228246

(71)Applicant : HITACHI LTD
HITACHI INFORMATION
TECHNOLOGY CO LTD

(22)Date of filing : 12.08.1999

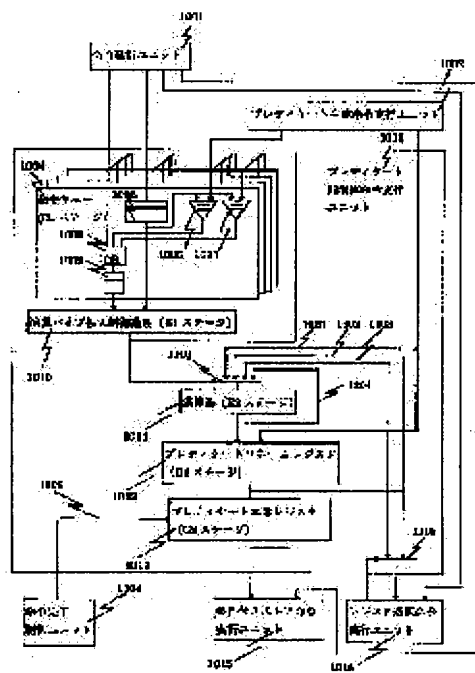
(72)Inventor : TOKORO MASAHIRO
MORIYAMA YOSHIYA

(54) OUT-OF-ORDER EXECUTION SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To shorten the process time from the generation of a predicate to conditioned execution by providing a dedicated inter-predicate operation unit which performs out-of-order execution.

SOLUTION: When it is judged that operation pipe feed control logic 1010 can not execute an instruction (register 1009 is ineffective), no instruction is fed to an operation pipe. In this case, a comparator 1006 compares a rename register number enqueued an instruction queue 1004 with a rename register number generated by a predicate generation instruction execution unit 1002 and the register 1009 is made effective through an OR circuit 1008 when they match each other. Consequently, the operation pipe feed control logic 1010 feeds an instruction to the operation pipe. Source data of operation are sent out to a computing element 1011 by selecting a forwarding bus 1201 from a predicate generation instruction execution unit 1002 by a selector 1101 and operated.



【特許請求の範囲】

【請求項1】プレディケートと呼ばれる真偽値による実行条件を命令文中に保持し、プレディケート値が真の場合にだけその命令を実行（条件付き実行）する機構を備えるコンピュータシステムにおいて、実行命令文中の複数のプレディケート値間の論理演算（プレディケート間演算）を行う演算器を浮動小数点演算や整数演算からプレディケート値を生成する演算器とは別個に設け、その演算器で実行される命令と他の演算器で実行される命令の実行順序が、実行命令文中の命令順序によらないことを特徴としたアウトオブオーダー実行方式。

【請求項2】プレディケートと呼ばれる真偽値による実行条件を命令文中に保持し、プレディケート値が真の場合にだけその命令を実行（条件付き実行）する機構を備えるコンピュータシステムにおいて、実行命令文中の複数のプレディケート値間の論理演算（プレディケート間演算）を行う演算器を浮動小数点演算や整数演算からプレディケート値を生成する演算器とは別個に設け、その演算器で実行される命令と他の演算器で実行される命令の実行順序が、実行命令文中の命令順序によらず、かつプレディケートデータ間の演算を行う複数の命令の間の実行順序も実行命令文中の命令順序によらないことを特徴としたアウトオブオーダー実行方式。

【請求項3】請求項1または2のアウトオブオーダー実行方式において、

プレディケート間演算命令を実行するユニットが、命令キュー、演算器、プレディケート間演算結果を一時的に保持しておくレジスタ（プレディケートリネームレジスタ）、および実行命令文順序（インオーダー）保証後のプレディケート値を格納するレジスタ（プレディケート

本体レジスタ）を所有し、プレディケート間演算命令を発行するユニットが、演算リソースとして、命令コード、演算に用いるプレディケートデータ（ソースデータ）が格納されているプレディケート本体レジスタ番号及びプレディケートリネームレジスタ番号、有効なソースデータが本体レジスタ、リネームレジスタのどちらに格納されているかを示すフラグ、レジスタに格納されているソースデータが有効または無効であることを示すフラグを発行し、

プレディケート間演算命令実行ユニットはそれらの演算リソースを命令キューへ格納し、命令発行ユニットが演算リソースを発行した時点でソースデータが全て有効な命令は、命令キュー内で実行可能状態へ遷移させ、また命令発行ユニットが演算リソースを発行した時点でソースデータの中に無効なものを含む場合、自演算器および他演算器が生成したプレディケートデータが格納されるリネームレジスタ番号を迂回経路（バイパス）を介して受け取る（フォワーディング）ことで実行可能状態へ遷移させ、

命令キューに格納した演算リソース中の実行可能なもの

10

20

30

40

50

から、プレディケートリネームレジスタおよびプレディケート本体レジスタより演算ソースデータを読み出してプレディケート間演算を実行し、演算結果を一時プレディケートリネームレジスタへ格納し、その後実行命令文順序を保証する命令完了制御ユニットからの指示によりプレディケート本体レジスタへ格納することを特徴としたアウトオブオーダー実行方式。

【請求項4】請求項1または2のアウトオブオーダー実行方式において、

プレディケート間演算命令を実行するユニットが、リザベーションステーション、演算器、プレディケート間演算結果を一時的に保持しておくレジスタ（プレディケートリネームレジスタ）、および実行命令文順序（インオーダー）保証後のプレディケートを格納するレジスタ

（プレディケート本体レジスタ）を所有し、プレディケート間演算命令を発行するユニットが、演算リソースとして、命令コード、ソースデータが格納されるプレディケートリネームレジスタ番号、演算ソースプレディケートデータ、演算ソースデータが有効または無効であることを示すフラグを発行し、それらの演算リソースをリザベーションステーションとよぶ命令キューへ格納し、

命令発行ユニットが演算リソースを発行した時点でソースデータが全て有効な場合は、リザベーションステーション内で実行可能状態へ遷移させ、

命令発行ユニットが演算リソースを発行した時点でソースデータの中に無効なものを含む場合、自演算器および他演算器が生成したプレディケートデータが格納されるリネームレジスタ番号および演算ソースであるプレディケートデータを迂回経路（バイパス）を介して受け取る（フォワーディング）ことで実行可能状態へ遷移させ、リザベーションステーションに格納した演算リソース中の実行可能なものから、リザベーションステーションに格納したソースデータを使用してプレディケート間演算を実行し、演算結果を一時プレディケートリネームレジスタへ格納し、その後実行命令文順序を保証する命令完了制御ユニットからの指示によりプレディケート本体レジスタへ格納することを特徴としたアウトオブオーダー実行方式。

【請求項5】請求項3および請求項4のプレディケート間演算命令のアウトオブオーダー実行方式を用いたプロセッサ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明はプレディケートを用いる条件付き実行方式とその実行方式を用いたプロセッサに関する。

【0002】

【従来の技術】スーパースカラやVLIW（Very Long Instruction Word）など、命令レベルの並列化を目的と

するコンピュータにおいて、プレディケートと呼ばれる真偽値を実行条件として命令中に保持し、プレディケート値が真の場合にだけその命令を実行（条件付き実行）する機構を持つものがある。

【0003】このプレディケートにおいてコンパイラは命令レベルのスケジューリングなどを用いて、高速な実行コードを生成するために、プレディケートデータを生成する命令（プレディケート生成命令）、実行命令文中の2つのプレディケートデータ間の包含関係の演算を行う命令（プレディケート間演算命令）、プレディケート値が真の場合にだけメモリにストア動作を行う命令（条件付きストア命令）、プレディケート値が真の場合と偽の場合とで違うレジスタの値を選択する命令（レジスタ選択命令）を生成する。

【0004】複数の条件に基づきストアまたはレジスタの選択を行う場合、プレディケート生成命令で生成された複数のプレディケートデータをもとにプレディケート間演算命令で条件の真偽値を求め、その値を用いて条件付きストア命令またはレジスタ選択命令を実行する。

【0005】プロセッサは、プレディケート生成命令実行ユニットが生成したプレディケートデータをプレディケート間演算命令実行ユニットに送出する。プレディケート間演算命令実行ユニットは、実行文中の命令順序に従い演算を行い、プレディケート値が確定した後に、条件付きストア命令実行ユニット及びレジスタ選択命令実行ユニットにプレディケートデータを送出する。条件付きストア命令実行ユニット及びレジスタ選択命令実行ユニットはこのプレディケート値を評価し条件付き実行を行う。

【0006】図2に従来のプレディケート技術を用いた条件付き実行（レジスタ選択命令の実行）のタイムチャートの例を示す。

【0007】図2では実行命令文順に命令1から命令5の5命令が実行される場合のタイムチャートを示した。命令1がプレディケート生成命令、命令2～4がプレディケート間演算命令、命令5がレジスタ選択命令であり、命令2が命令1演算結果を、命令4が命令2と3の演算結果を、命令5が命令4の演算結果をそれぞれソースデータとして用いる。その他のプレディケートソースデータにはプレディケート本体レジスタの値を用いる。（命令発行時にプレディケートレジスタ内に有効なデータが存在する。）

プレディケート生成命令、プレディケート間演算命令、レジスタ選択命令はともに命令発行後5サイクルで終了すると仮定し、それらのサイクルをそれぞれ、D、E0、E1、E2、Cステージと呼ぶ。Dステージで各命令実行ユニットへの命令の発行、E0ステージで各実行ユニットへの命令の取り込み、E1ステージで演算器への命令投入の判定、E2ステージで演算、Cステージでレジスタへの実行結果の書き込みが行われるものとす

る。

【0008】また、プレディケート生成命令実行ユニット、プレディケート間演算命令実行ユニット、レジスタ選択命令実行ユニットで同時に処理できる命令数はともに1であり、パイプライン処理できるものと仮定した。プレディケート生成命令実行ユニットの演算器出力とプレディケート間演算命令実行ユニットの演算器入力の間およびプレディケート間演算命令実行ユニットの演算器の出力ー入力間にE2ステージの演算結果を次サイクルのE2演算ステージに反映させるためのバイパスの存在も仮定した。

【0009】まず、命令発行ユニットが2000のタイミングで命令1をプレディケート生成命令実行ユニットへ発行する。プレディケート生成命令実行ユニットは、2001のタイミングで命令1を取り込む。命令1のソースデータは有効であるため2002のタイミングで命令の実行が可能であること判断し命令実行信号を演算パイプへ送出する。演算パイプは2003のタイミングでプレディケートデータの生成を行い、生成したデータを2004のタイミングでプレディケートレジスタへ格納する。

【0010】命令2は命令1の1サイクル後2013のタイミングでプレディケート間演算命令実行ユニットの演算器で演算が行われる。命令1が生成する命令2のソースデータはプレディケート生成命令実行ユニットの演算器からプレディケート間演算命令実行ユニットの演算器へのバイパスを用いてフォワーディングされる。2012のタイミングでの演算器への命令投入決定を可能にするため、プレディケート間演算命令実行ユニットはプレディケート生成命令実行ユニットから2002のタイミングで命令1の命令実行を示す信号と演算結果の書き込み先レジスタ番号を受け取る。

【0011】プレディケート間演算命令実行ユニットのパイプがパイプライン化されているため、命令2の1サイクル後2023のタイミングで命令3が、さらに1サイクル後の2033のタイミングで命令4の演算が行われる。命令4の演算は命令2と命令3の演算結果をソースデータとして用いるが、命令2の生成データはプレディケートレジスタから、命令3の生成データはバイパスから読み出す。

【0012】レジスタ選択命令5は、命令4が生成するデータがプレディケートレジスタに書き込まれた後に2043のタイミングでその値を読み出しプレディケート値に従いレジスタの選択を行う。

【0013】

【発明が解決しようとする課題】プレディケートの生成から条件付き実行に至るまでの処理時間の短縮を目的とする。

【0014】

【課題を解決するための手段】プレディケート間演算と

呼ぶ実行命令文中の2つのプレディケート間の包含関係の演算を行う演算器を個別に設け、その演算器が、他演算器が生成したプレディケートデータをフォワーディングしながらアウトオブオーダー実行することにより条件付き実行の処理を高速化する。

【0015】

【発明の実施の形態】以下、本発明に基づくプレディケート間演算命令のアウトオブオーダー実行の一実施例を図面により詳細に説明する。

【0016】図1はプレディケート間演算命令のアウトオブオーダー実行を行うハードウェアの構成を示すブロック図である。

【0017】命令発行ユニット1001はプレディケート間演算命令実行ユニット1003へプレディケート間演算命令情報を発行する。この時命令発行ユニット1001はプレディケート生成命令実行ユニット1002、条件付きストア命令実行ユニット1015、およびレジスタ選択命令実行ユニット1016へプレディケート生成命令情報、レジスタ選択命令情報、条件付きストア命令情報をそれぞれ同時に発行しても良い。

【0018】発行されたプレディケート間演算命令情報は、命令キュー1004内のレジスタ1005とレジスタ1009に格納する。レジスタ1009にはOR回路1008を介して、発行した命令のソースデータが有効である（命令のソースデータがプレディケートリネームレジスタ1012またはプレディケート本体レジスタ1013のいずれかに存在する）ことを示すフラグを格納する。

【0019】プレディケート間演算命令のソースデータが複数ある場合、レジスタ1009は各レジスタ対応にソースデータの数だけ存在する。

【0020】命令キュー1004に格納した情報は、レジスタ1009が全て有効である場合、演算パイプ投入制御論理1010が命令の実行が可能な状態と判断し演算パイプへ命令の投入を行う。

【0021】演算パイプと投入制御論理1010は命令キュー1004に格納されている複数の命令のうち、実行可能なものを演算パイプへ投入するが、投入の順序は実行命令文順序（プログラムオーダー）に依存する必要はない。

【0022】命令発行時点でレジスタ1009が有効な場合は、演算のソースデータがプレディケートリネームレジスタ1012かプレディケート本体レジスタ1013のいずれかに格納されているので、読み出しパス1202か読み出しパス1203のいずれかから、セクタ1101により選択し演算器1011へ送出し演算する。

【0023】演算パイプ投入制御論理1010が命令の実行が不可能な状態と判断した場合（レジスタ1009が無効である場合）は、演算器パイプへ命令の投入は行

わない。この場合命令キュー1004は、プレディケート生成命令実行ユニット1002、およびプレディケート間演算命令実行ユニット1003の演算結果を待ち合わせている状態になる。

【0024】命令キュー1004内に待ち合わせているリネームレジスタ番号とプレディケート生成命令実行ユニット1002が生成したリネームレジスタ番号はコンパレータ1006により比較され、一致した場合は、OR回路1008を介して、レジスタ1009を有効にする。これにより演算パイプ投入制御論理1010は演算パイプへ命令の投入を行う。

【0025】演算のソースデータは、プレディケート生成命令実行ユニット1002からのフォワーディングパス1201をセクタ1101により選択し、演算器1011へ送出し演算する。またプレディケート生成命令実行ユニット1002が生成したプレディケートデータは、プレディケートリネームレジスタ1012へ格納する。

【0026】命令キュー1004内に待ち合わせているリネームレジスタ番号とプレディケート間演算命令実行ユニット1003が実行した演算結果を格納するリネームレジスタ番号はコンパレータ1007により比較し、一致した番号は、OR回路1008を介して、レジスタ1009を有効にする。これにより演算パイプ投入制御論理1010は演算パイプへ命令の投入を行う。

【0027】演算のソースデータは、演算器1011の演算結果のフォワーディングパス1204をセクタ1101により選択し、演算器1011へ送出し演算する。

【0028】演算結果はプレディケートリネームレジスタ1012へ格納する。

【0029】プレディケートリネームレジスタ1012に格納した演算結果、およびプレディケート生成命令実行ユニットが生成したプレディケートデータは、命令完了制御ユニット1014が命令実行順序を保証してプレディケート本体レジスタ1013へライトバックする。

【0030】条件付きストア命令実行ユニット1015はプレディケート本体レジスタ1013に確定したプレディケートを使用し、条件付きストア命令実行ユニット1017内でプレディケート値を評価し、プレディケート値が真の場合にだけメモリにストア動作を行うことにより条件付き実行を行う。

【0031】レジスタ選択命令実行ユニット1016は、命令発行ユニットが発行するプレディケートリネームレジスタ番号、および本体レジスタ番号を受け取り、必要なプレディケートデータをセクタ1102により選択し、プレディケート値が真の場合と偽の場合とで違うレジスタの値を選択することで条件付き実行を行う。

【0032】図3に本発明によるプレディケートを用いた条件付き実行（レジスタ選択命令を実行）のタイムチ

ャートを示す。

【0033】図2の場合と同一の命令例を実行し、図2の時と同様に命令発行後5サイクルで演算結果をレジスタへ書き込むと仮定し、それらのサイクルをそれぞれ、D、E0、E1、E2、C1ステージと呼ぶ。Dステージで各命令実行ユニットへの命令の発行、E0ステージで各実行ユニットへの命令の取り込み、E1ステージで演算器への命令投入の判定、E2ステージ演算、C1ステージでリネームレジスタへの実行結果の書き込みが行われる。その後C2ステージと呼ばれるサイクルでリネームレジスタから本体レジスタへのライトバックを行う。

【0034】命令発行ユニットは同時に4命令を発行できるものと仮定した。また、プレディケート生成命令実行ユニット、プレディケート間演算命令実行ユニット、レジスタ選択命令実行ユニットで同時に処理できる命令数はともに1であり、パイプライン処理できるものと仮定した。

【0035】命令発行ユニットはプレディケート生成命令実行ユニット、プレディケート間演算命令実行ユニットに同タイミングで命令1～4の命令情報をそれぞれ3000、3010、3020、3030のタイミングにより発行を行う。

【0036】プレディケート生成命令実行ユニットは3000のタイミングで発行された命令1の命令情報を3001のタイミングで取り込む。命令1のソースデータは命令発行時にすでに有効であるので、3002のタイミングで実行が可能であると判断し、命令実行信号を演算パイプへ送出する。演算パイプは3003のタイミングでプレディケートの生成を行い、3004のタイミングでリネームレジスタへの書き込みを行う。プレディケートリネームレジスタに書き込まれた演算結果は3005のタイミングでプレディケート本体レジスタにライトバックされる。

【0037】プレディケート間演算命令実行ユニットは3010、3020、3030のタイミングで発行された命令2、命令3、命令4の命令情報を3011、3021、3031のタイミングで取り込む。

【0038】命令2はソースデータとして命令1の演算結果を用いるが、命令情報を命令キューに取り込んだ時点ではソースデータが無効であるため、E0ステージでストールする。プレディケート間演算命令実行ユニットはプレディケート生成命令実行ユニットから3002のタイミングで命令1の命令実行信号と演算結果の書き込み先プレディケートリネームレジスタ番号を受け取り、命令2の状態を実行不可能状態から実行可能状態へ遷移させる。命令2は3012のタイミングで実行可能であると判断され、3013のタイミングで演算、3014のタイミングでリネームレジスタへの書き込み、3015のタイミングでプレディケート本体レジスタへのライ

トバックが行われる。

【0039】命令3は命令2と同時にプレディケート間演算命令実行ユニットの命令キューに取り込まれるが、命令キューに取り込まれた時点でソースデータが有効であるため、E0ステージでストールせず命令2を追い越して3022のタイミングで演算器への投入が行われる（アウトオブオーダー実行）。3023のタイミングで演算、3024のタイミングでプレディケートリネームレジスタへの書き込みが行われるが、先行する命令2のライトバック（3015）を待つためC1ステージでストールする。プレディケート本体レジスタへのライトバックは命令2と同時の3025のタイミングで行われるが、書き込み時に命令2のデータを命令3のデータで上書きすることにより、プレディケート本体レジスタのインオーダー保証を行う。

【0040】命令4はソースデータとして命令2と命令3の演算結果を用いる。命令発行時点ではソースデータが無効であるため、2つのソースデータが共に有効になるまでE0ステージでストールする。プレディケート間演算命令実行ユニットでは3012、3022のタイミングでそれぞれ命令2と命令3の命令実行信号、演算結果書き込み先プレディケートリネームレジスタ番号をもとに命令4のソースデータが有効であることを示すレジスタの値を真にする。それにより、命令4は3032のタイミングで実行可能となり、3033のタイミングで演算、3034のタイミングで演算結果のプレディケートリネームレジスタへの書き込み、3035のタイミングでプレディケート本体レジスタへのライトバックを行う。命令5は3040のタイミングで命令発行ユニットから発行され、3041のタイミングでレジスタ選択命令実行ユニットの命令キューに取り込まれる。命令5はソースデータとして用いる命令4の演算結果を待つためにE0ステージでストールし、3042のタイミングで演算器への投入、3043でプレディケートリネームレジスタからプレディケート値を読み出し、その値に基づいてレジスタ選択の実行が行われる。

【0041】

【発明の効果】以上説明した様に、図2で示した従来の方式では命令1の発行から命令5がレジスタの選択を実行するまでに必要とするサイクル数は9サイクルであるのに対し、本発明を用いた図3では8サイクルで一連の処理が終了する。

【0042】この様に、アウトオブオーダー実行を行う専用のプレディケート間演算ユニットを設けることにより、プレディケートを用いた条件付き実行を行うプロセッサの性能を向上させることができる。

【図面の簡単な説明】

【図1】本発明によるプレディケート間演算実行ユニットの構成例である。

【図2】従来のプレディケートを用いた条件付き実行

10

20

30

40

50

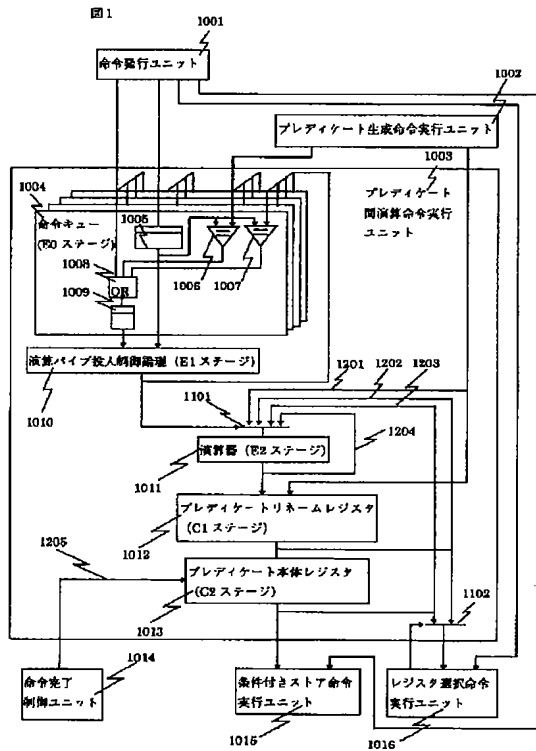
(レジスタ選択命令の実行)のタイムチャートである。

【図3】本発明によるプレディケートを用いた条件付き実行(レジスタ選択命令の実行)のタイムチャートである。

【符号の説明】

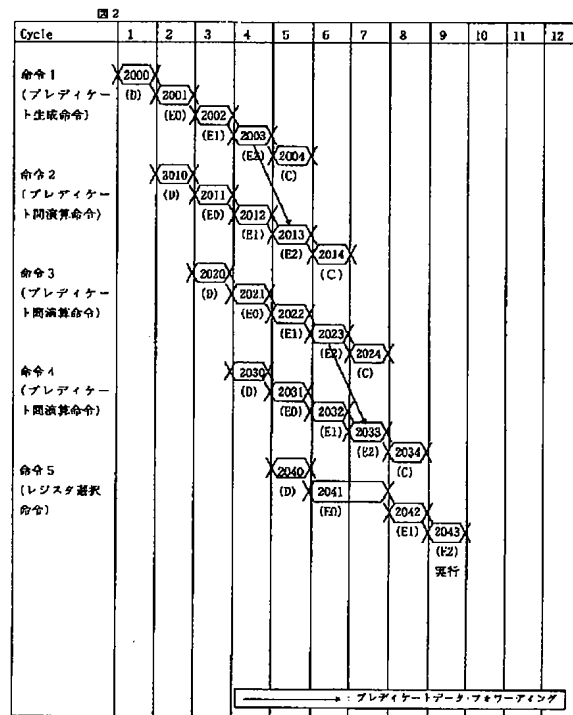
1001…命令発行ユニット、1002…プレディケート生成命令実行ユニット、1003…プレディケート間演算命令実行ユニット、1004…命令キュー、1005…命令情報保持ラッチ、1006…リネームレジスタ番号比較器、1007…リネームレジスタ番号比較器、1008…OR回路、1009…ソースデータ有効フラグ保持ラッチ、1010…演算パイプ投入制御論理、1*

【図1】

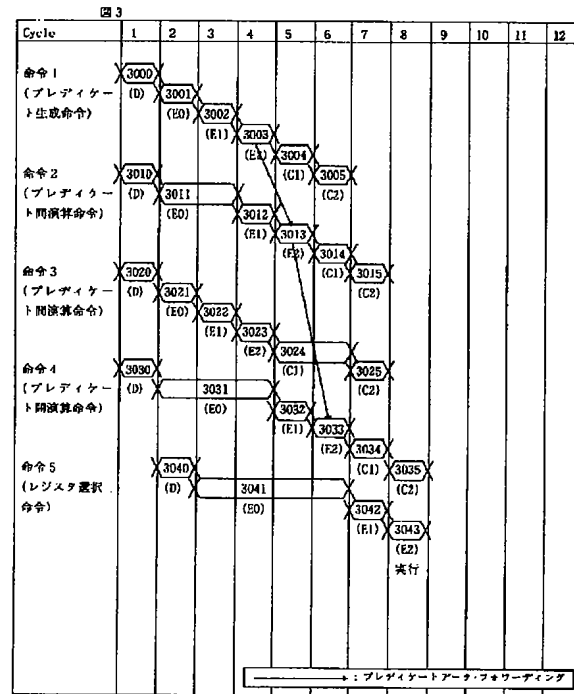


*011…演算器、1012…プレディケートリネームレジスタ、1013…プレディケート本体レジスタ、1014…命令完了制御ユニット、1015…条件付きストア命令実行ユニット、1016…レジスタ選択命令実行ユニット、1101…プレディケート間演算ソースデータセクタ、1102…レジスタ選択ソースデータセクタ、1201…プレディケート生成命令実行結果バイパス、1202…プレディケートリネームレジスタ読み出しパス、1203…プレディケート本体レジスタ読み出しパス、1204…プレディケート間演算命令実行結果バイパス、1205…命令完了制御信号。

【図2】



【図3】



フロントページの続き

(72)発明者 森山 善哉
神奈川県秦野市堀山下1番地 株式会社日
立インフォメーションテクノロジー内

Fターム(参考) 5B013 AA12 BB04 CC10
5B033 AA14 BD00 CA22